

## **Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of the claims in the application.

1. (Currently amended) A method, comprising:

(A) obtaining, at a server, a plurality of e-mails from one or more remote client computers, wherein the plurality of e-mails are intended for distribution by the server to a plurality of respective remote destinations;

(B) creating, using one or more suitably programmed computers, a respective data node for each respective e-mail in said plurality of e-mails, wherein each respective data node includes (i) a pointer, wherein the pointer identifies to the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks a number of attempts made to deliver the respective e-mail;

(C) processing, using one or more suitably programmed computers, the plurality of data nodes solely within a non-persistent storage comprising a plurality of queues, wherein each queue in the plurality of queues corresponds to a specific domain, without requiring that information indicative of the e-mails be written to and then read from persistent storage during the processing (C) of the data nodes, wherein said processing (C) comprises, for each respective data node:

(i) determining a destination domain of the respective data node;

(ii) adding the respective data node to a queue in the non-persistent storage corresponding to the destination domain of the respective data node when the queue exists in the plurality of queues; and

(iii) creating a queue in the non-persistent storage corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist in the plurality of queues, wherein

said processing (C) further comprises a method comprising:

(a) selecting a first respective queue in the plurality of queues that contains data nodes;

(b) retrieving e-mails corresponding to each of the data nodes in the first respective queue;

(c) finding a remote server corresponding to the destination domain of the respective queue;

(d) sending each of the retrieved e-mails corresponding to each of the data nodes in the first respective queue to the remote server corresponding to the [[a]] destination domain of the first respective queue, wherein the sending (d) comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the respective queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail,

wherein, when a delivery failure message is received for a first e-mail in the retrieved e-mails at a time after said sending (d), said processing (C) further comprises:

(i) pushing the data node corresponding to the first e-mail back onto the respective queue, and

(ii) incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the first e-mail;

and

(e) extinguishing the first respective queue when all e-mails in the respective queue have been delivered without receiving a delivery failure message.

2. (Previously presented) A method as in claim 1, further comprising

storing, in persistent storage, recovery information indicative of the processing, said recovery information being used for recovery from a system fault.

3. (Previously presented) A method as in claim 2, wherein said recovery information includes information indicative of the plurality of e-mails, wherein said information is indicative of less than the entirety of each e-mail in said plurality of e-mails.

4. (Previously presented) A method as in claim 3, wherein said information indicative of an e-mail in the plurality of e-mails includes a bit vector.

5. (Currently amended) A method as in claim 1, wherein said sending (d) of each of the e-mails corresponding to each of the data nodes in the first queue from the first queue to the destination domain is done at a specific sending instance.

6. (Currently amended) A method as in claim 5, wherein said sending (d) comprises opening a communication channel to a single specified domain and sending each of the e-mails within the single communication channel.

7. (Previously presented) A method as in claim 3, wherein said recovery information includes a numerical designation for each e-mail in said plurality of e-mails, and a state of processing of each e-mail in said plurality of e-mails.

8. (Cancelled)

9. (Currently amended) A method as in claim 1, wherein said selecting (a) comprises selecting a queue in the plurality of queues which has the greatest number of the e-mails within the queue.

10. (Currently amended) A method as in claim 1, wherein said selecting (a) comprises selecting a queue in the plurality of queues which has existed for the greatest period of time.

11. (Currently amended) A method as in claim 1, further comprising, during said ~~selection~~ selecting of said first queue (a), asynchronously looking up domain name server information for a second queue in the plurality of queues, different than the first queue, and selecting the second queue.

12. (Currently amended) A method as in claim 1, wherein the creating ~~step~~ (B) separates personalized information about each e-mail in the plurality of e-mails from non-personalized information.

13. (Previously presented) A method as in claim 12, wherein said non-personalized information includes e-mail destination information.

14. (Previously presented) A method as in claim 5, wherein said processing further comprises:  
determining information about processing by said destination domain; and  
adjusting a speed of sending of the e-mails based on said information about processing of said destination domain.

15. (Previously presented) A method as in claim 14, wherein said information about processing comprises a speed of e-mail processing.

16. (Previously presented) A method as in claim 1, further comprising:  
maintaining a log representing information relating to a number of e-mails in said plurality of e-mails which have been processed; and  
comparing contents of said log with licensing information, to determine if the number of e-mails that has been processed exceeds a licensed number.

17. (Currently amended) A method as in claim 1, comprising:  
storing recovery information about a state of processing of the plurality of e-mails to persistent storage, wherein said recovery information comprises less than the entirety of each of the e-mails in the plurality of e-mails; and  
wherein the processing (C) of the plurality of e-mails directs the plurality of e-mails to a desired location without writing the plurality of e-mails to persistent storage during said processing.

18-19. (Cancelled)

20. (Previously presented) A method as in claim 17, wherein said recovery information includes information indicative of said plurality of e-mails, wherein said information is indicative of less than the entirety of each e-mail in said plurality of e-mails.

21. (Cancelled)

22. (Currently amended) A method as in claim 17, wherein said processing (C) comprises:

arranging information about the e-mails into [[a]] said plurality of queues, ~~each queue in said plurality of queues representing a single domain~~; and

sending e-mails to a recipient, by sending a plurality of e-mails to a single domain, represented by a queue in said plurality of queues, at a specific sending instance.

23. (Cancelled)

24. (Previously presented) A method as in claim 17, wherein said recovery information includes a number of e-mails, and a state of processing of each e-mail in said number of e-mails.

25-28. (Cancelled)

29. (Previously presented) A method as in claim 17, further comprising:

processing the plurality of e-mails by separating personalized information about each e-mail in the plurality of e-mails from non-personalized information.

30. (Previously presented) A method as in claim 29, wherein said non-personalized information includes destination information for the plurality of e-mails.

31. (Previously presented) A method as in claim 22, wherein said processing comprises:

determining a speed of processing of said single domain; and

adjusting a speed of sending of the e-mails based on said speed of processing of said single domain.

32. (Previously presented) A method as in claim 17, further comprising:

maintaining a log representing information relating to e-mails which have been processed; and

comparing contents of said log with licensing information, to determine if said information relating to e-mails exceeds a licensed number.

33-83. (Cancelled)

84. (Currently amended) A computer system comprising:

(A) means for obtaining a plurality of e-mails from one or more remote client computers, wherein the plurality of e-mails are intended for distribution by the means for obtaining to a plurality of respective remote destinations;

(B) means for creating a respective data node for each respective e-mail in said plurality of e-mails, wherein each respective data node includes (i) a pointer, wherein the pointer identifies to the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks a number of attempts made to deliver the respective e-mail;

(C) means for processing the plurality of data nodes ~~solely within a non-persistent non-persistent storage comprising a plurality of queues, wherein each queue in the plurality of queues corresponds to a specific domain,~~ without requiring that information indicative of the e-mails be written to and then read from persistent storage during the processing of the e-mails, wherein said processing (C) comprises, for each respective data node:

(i) determining a destination domain of the respective data node; ~~and~~

(ii) adding the respective data node to a queue in the non-persistent storage corresponding to the destination domain of the respective data node when the queue exists in the plurality of queues; ~~and~~

(iii) creating a queue in the non-persistent storage corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist in the plurality of queues; and wherein

the means for processing (C) further comprises instructions for performing a method comprising:

(a) selecting a ~~first~~ respective queue in the plurality of queues that contains data nodes;

(b) retrieving e-mails corresponding to each of the data nodes in the ~~first~~ respective queue;

(c) finding a remote server corresponding to the destination domain of the respective queue;

(d) sending each of the retrieved e-mails corresponding to each of the data nodes in the ~~first~~ respective queue to the remote server corresponding to the [[a]] destination domain of the ~~first~~ respective queue, wherein the sending (d) comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the respective queue and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail,

wherein, when a delivery message is received for a first e-mail in the retrieved e-mails at a time after the sending (d), said means for processing (C) further comprises instruction for:

(i) pushing the data node corresponding to the first e-mail back onto the respective queue, and

(ii) incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the respective e-mail; and

(e) extinguishing the ~~first~~ respective queue when all e-mails in the respective queue have been delivered without receiving a delivery failure message.

85-86. (Cancelled)

87. (New) A computer system comprising:

(A) a module for obtaining a plurality of e-mails from one or more remote client computers, wherein the plurality of e-mails are intended for distribution to a plurality of respective remote destinations;

(B) a module for creating a respective data node for each respective e-mail in said plurality of e-mails, wherein each respective data node includes (i) a pointer, wherein the pointer identifies the corresponding respective e-mail in persistent storage, (ii) an identification of the recipient of the respective e-mail, (iii) an identification of the sender of the respective e-mail, (iv) a destination domain of the respective e-mail, and (v) a visit count that tracks a number of attempts made to deliver the respective e-mail;

(C) a module for processing the plurality of data nodes ~~solely~~ within a non-persistent storage comprising a plurality of queues, wherein each queue in the plurality of queues corresponds to a specific domain, without requiring that information indicative of the e-mails be written to and then read from persistent storage during the processing of the e-mails, wherein said processing (C) comprises, for each respective data node:

(i) determining a destination domain of the respective data node;

(ii) adding the respective data node to a queue in the non-persistent storage corresponding to the destination domain of the respective data node when the queue exists in the plurality of queues;

(iii) creating a queue in the non-persistent storage corresponding to the destination domain and adding the respective data node to the created queue when the queue does not exist in the plurality of queues; and wherein

the processing module (C) further comprises instructions for performing a method comprising:

(a) selecting a respective queue in the plurality of queues that contains data nodes;

(b) retrieving e-mails corresponding to each of the data nodes in the respective queue;

(c) finding a remote server corresponding to the destination domain of the respective queue;

(d) sending each of the retrieved e-mails corresponding to each of the data nodes in the respective queue to the remote server corresponding to the destination domain of the respective queue, wherein the sending (d) comprises reconstructing an e-mail in the retrieved e-mails from (i) the data node corresponding to the e-mail in the respective queue



and (ii) the e-mail in persistent storage identified by the pointer to the respective e-mail that is in the data node corresponding to the e-mail,

wherein, when a delivery message is received for a first e-mail in the retrieved e-mails at a time after the sending (d), said module for processing (C) further comprises instructions for:

(i) pushing the data node corresponding to the first e-mail back onto the respective queue, and

(ii) incrementing the visit count in the data node corresponding to the first e-mail to account for the failed delivery of the respective e-mail; and

(e) extinguishing the respective queue when all e-mails in the respective queue have been delivered without receiving a delivery failure message.

88. (New) The method of claim 1, wherein the processing (C) further comprises:

creating a unique identifier for the e-mail corresponding to the respective node from a bit vector of the respective data node stored in the queue, and

storing the bit vector in persistent storage, wherein the bit vector comprises sufficient information to reconstruct the corresponding e-mail and the state of processing of the corresponding e-mail.

89. (New) The computer system of claim 84, wherein the means for processing (C) further comprises:

creating a unique identifier for the e-mail corresponding to the respective node from a bit vector of the respective data node stored in the queue, and

storing the bit vector in persistent storage, wherein the bit vector comprises sufficient information to reconstruct the corresponding e-mail and the state of processing of the corresponding e-mail.

90. (New) The computer system of claim 87, wherein the means for processing (C) further comprises:

creating a unique identifier for the e-mail corresponding to the respective node from a bit vector of the respective data node stored in the queue, and

storing the bit vector in persistent storage, wherein the bit vector comprises sufficient information to reconstruct the corresponding e-mail and the state of processing of the corresponding e-mail.

91. (New) The method of claim 1, wherein the method comprising steps (a) through (e) of the processing (C) is concurrently performed for each queue in the plurality of queues.

92. (New) The computer system of claim 84, wherein the instructions for performing the method comprising steps (a) through (e) of the means for processing (C) is concurrently performed for each queue in the plurality of queues.

93. (New) The computer system of claim 87, wherein the instructions for performing the method comprising steps (a) through (e) of the module for processing (C) is concurrently performed for each queue in the plurality of queues.

94. (New) The method of claim 1, wherein two or more queues in the plurality of queues correspond to the same destination domain.

95. (New) The computer system of claim 84, wherein two or more queues in the plurality of queues correspond to the same destination domain.

96. (New) The computer system of claim 87, wherein two or more queues in the plurality of queues correspond to the same destination domain.

97. (New) The method of claim 1, wherein each queue in the plurality of queues corresponds to a different destination domain.

98. (New) The computer system of claim 84, wherein each queue in the plurality of queues corresponds to a different destination domain.

99. (New) The computer system of claim 87, wherein each queue in the plurality of queues corresponds to a different destination domain.